Participatory Font Democracy: A Development Report

by Edward H. Trager
2009.01.06

What is the most important innovation in typography in the past 25 years? Answers will vary, but most people will probably cite technological innovations such as the emergence of the PostScript, TrueType and OpenType scalable font technologies [1], or perhaps mention the coming of age of the Unicode standard originally proposed by Joseph Becker, Lee Collins, and Mark Davis back in 1988 [2]. Others might cite the emergence of the World Wide Web in 1991 [3].

Asking this question today, few would be likely to list *participatory democracy* as an important factor in modern digital typography. However, when we look back on the history of typography in five or ten years, I believe this term will rank highly.

In the past few years, an increasing number of Open Source Unicode fonts of very high quality for specific scripts and specific uses (*scholarly*, *scientific*, etc.) have been released by dedicated groups and individuals for the common benefit. The emergence of George Williams' Open Source outline font editor, FontForge [4], has certainly played an important role in lowering the barrier to entry, enabling amateurs, students, and professional designers alike to design digital type. SIL's Open Font License [5], developed in 2005, has already had a noticeable impact in facilitating the development of an open typographic community by providing a legal framework and infrastructure for the worldwide development, sharing and improvement of fonts. One need only take a cursory look at the excellent work being done by a wide spectrum of organizations —like SIL International [6], The Tibetan & Himalayan Digital Library [7], or the Khmer Software Initiative [8]— and individual people —like Christopher Harvey (Indigenous American language fonts at LanguageGeek.com [9]), Chris Fynn (Tibetan Jomolhari font [10]), or Fang Qianqian 房骞骞 (Wen Quan Yi 文泉驿 Chinese font [11])— to realize that *participatory font democracy* is a new and important force for freedom of expression in the modern world.

Many of us in the Open font development community are hoping that the Open Font Library (OFLB) [12] web site will become a premier hub for this new font democracy. To further this goal, I have participated by developing an AJAX-based font previewer which has been integrated into the new OFLB site currently under development.

The font previewer contains three major components:

1. FontPlayground is a Javascript-based client-side component that allows a user to type a phrase of his or her choosing and then see that phrase typeset in a selected font in real time.

2. PCFP is the server-based typesetter component which uses Pango and the Cairo graphics library to render text in a selected font to a bitmap PNG image which is sent back to the client. "PCFP" is abbreviated from "Pango-Cairo Font Playground."

3. KeyCurry is a Javascript-based client-side virtual keyboard component which allows the user to choose from over 100 keyboard mappings to facilitate typing not only in Latin and extended Latin, but also in most of the world's other major scripts.

Some commercial font vendors now offer similar AJAX-based font previewing capabilities. However, based on tests that I have conducted, FontPlayground is the *only* solution I know of which handles complex text layout correctly. Complex text layout (CTL) is required for proper shaping of scripts such as Arabic and Devanagari, and may be also required for extended Latin-based orthographies such as Vietnamese, African, and Indigenous American language orthographies.

Screenshots and details of the software components follow:

FontPlayground

The FontPlayground component as it currently appears on the OFLB development site.

The FontPlayground component uses a light-weight Javascript framework called Gladiator Components which I originally developed for creating "Web 2.0" bioinformatics applications where I work.

Integrating FontPlayground into a web page is simple. The `setListenersOnClass()` method of the `gFontPlayground` class adds an on click handler on images in an XHTML page that are tagged with a specific CSS class such as "typeSpecimen":

```
var myFP = new gFontPlayground("myFP");
myFP.setListenersOnClass("typeSpecimen");
```

Parameters such as the font family and the default text string to pass to the `pcfp` typesetter are simply placed in the `img` tag's `alt` property.

PCFP

The `pcfp` program is a server-side program written in C which uses Pango [13] and the Cairo graphics library [14] to typeset text to a PNG canvas. A simple PHP script provides the glue between the client and `pcfp`. Incorporation of Pango insures correct shaping behavior for complex scripts such as Arabic, Devanagari, and many others.

يولد جميع الناس أحرارًا متساوين في

An Arabic art font sample with correct shaping typeset by pcfp

KeyCurry

What does the Euro character "€" look like in Jos Buivenga's Fontin typeface? What does the German ess-tzet "ß" look like in Gentium? What does Hebrew Alef "א" look like in the Society for Biblical Studies (SBL) Hebrew font?

The ability to provide user-defined type specimens on-the-fly in near real-time in a web application has limited utility if users are unable or have difficulty typing the characters they want. To get around this problem, I developed KeyCurry.

KeyCurry is a client-side AJAX-based virtual keyboard component. As currently implemented on the OFLB development site, KeyCurry provides 123 distinct keyboard mappings ("keymaps") covering most of the world's major scripts, including Latin, extended Latin, Greek, Cyrillic, Arabic, Hebrew, Canadian Syllabics, Indic scripts, and many more.

The keymaps are adapted nearly unchanged from the Yudit Unicode text editor project by Gáspár Sinai [15]. KeyCurry provides an intuitive user interface for selecting the keymaps. Keymaps are loaded from the server using AJAX and then displayed in both a table and on a virtual keyboard. Appendix B provides a complete listing of currently available keymaps.

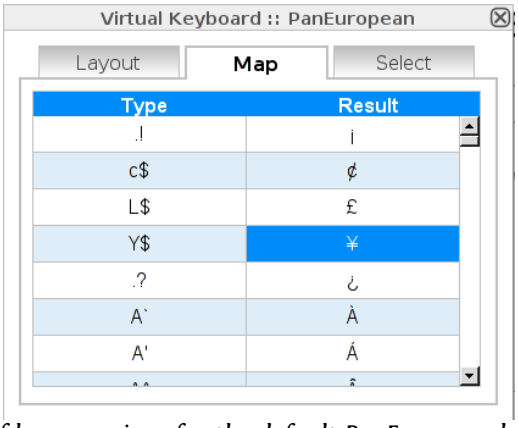
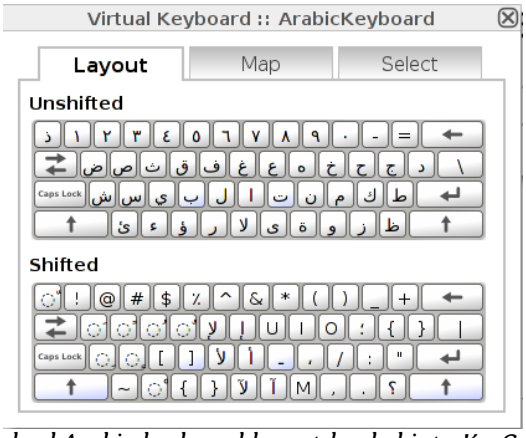


Table of key mappings for the default PanEuropean key map.



Standard Arabic keyboard layout loaded into KeyCurry.

References

1. The History of Fonts, http://www.prepressure.com/fonts/basics/history

2. “Unicode 88” by Joseph D. Becker Ph.D., Xerox Corp., Palo Alto, California, digital reprint available at http://www.unicode.org/history/unicode88.pdf.

3. http://en.wikipedia.org/wiki/Tim_Berners-Lee#Inventing_the_World_Wide_Web

4. FontForge, http://fontforge.sourceforge.net/

5. Open Font License, http://scripts.sil.org/OFL

6. SIL International, http://scripts.sil.org/

7. Tibetan Himalayan Digital Library, http://www.thdl.org/

8. Khmer Software Initiative, http://www.khmeros.info/

9. Language Geek, http://languagegeek.com/

10. Jomolhari, http://chris.fynn.googlepages.com/jomolhari

11. Wen Quan Yi, 文泉驿, http://wenq.org/enindex.cgi

12. Open Font Library, Development version staged at http://openfontlibrary.fontly.org/

13. Pango, http://www.pango.org/

14. Cairo Graphics, http://cairographics.org/

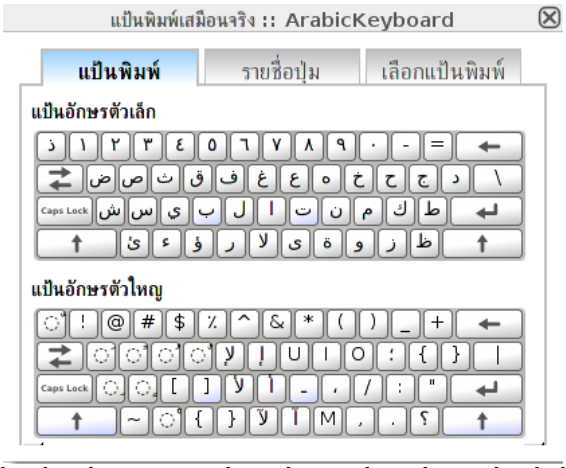
15. Yudit, http://www.yudit.org

Appendix A - Localization

Localization of the user interface components of FontPlayground is also easy to achieve by calling the `setLocale()` method of the global `LC` object prior to instantiating user interface objects:

```
LC.setLocale("zh-Hans");
var myFP = new gFontPlayground("myFP");
```

Message catalogs are in a simple JSON format. Message catalogs have been provided for French, Thai, Simplified Chinese and Traditional Chinese.



Thai localization (with Arabic Keyboard map loaded).

Appendix B - Currently Available Keymaps

ASCII-IPA.kmap American.kmap ArabTeX.kmap Arabic.kmap ArabicBuck.kmap ArabicKeyboard.kmap ArabicTranslit.kmap ArmenianEast.kmap ArmenianEastPhon.kmap ArmenianWest.kmap Azeri.kmap Baybayin.kmap Belarusian.kmap Bengali-Inscript.kmap Bengali.kmap BengaliSona.kmap Bulgarian-Translit.kmap Bulgarian.kmap CS-qwerty.kmap CS.kmap Cherokee.kmap Chinese-Pinyin.kmap Croatian.kmap Cyrillic.kmap Czech-deadkeys.kmap Czech.kmap Dakelh.kmap Danish.kmap Devanagari-Inscript.kmap Devanagari-Phonetic.kmap Devanagari-Velthuis.kmap Devanagari.kmap Dutch.kmap Esperanto.kmap Ethiopic.kmap Farsi.kmap French.kmap Georgian.kmap GeorgianB.kmap German.kmap Glagolitic.kmap	GrandLatin.kmap GreekAncient.kmap GreekBible.kmap GreekMonotonic.kmap GreekPolytonic.kmap Guarani.kmap Gujarati-Inscript.kmap Gujarati.kmap Gurmukhi-Inscript.kmap Gurmukhi.kmap Hanunoo.kmap Hebrew.kmap HebrewIsraeli.kmap Hungarian-prefix.kmap Hungarian.kmap HungarianRunes.kmap Inuktitut-ICI.kmap Inuktitut-KBD.kmap Israeli.kmap Kana.kmap Kannada-Inscript.kmap Kannada.kmap Kazakh-prefix.kmap Latin.kmap Lithuanian.kmap Malayalam-Inscript.kmap Malayalam.kmap MiddleKorean.kmap Mongolian.kmap Oriya-Inscript.kmap Oriya.kmap PaliRomanization.kmap PanEuropean.kmap Persian.kmap Polish-slash.kmap Polish.kmap Qwerty.kmap Romanian.kmap Runic-Futhark.kmap Runic-Futhorc.kmap Runic.kmap	Russian-ISO-Latinitsa.kmap Russian-JAVERTY.kmap Russian-Translit-German.kmap Russian-Translit-Nordic.kmap Russian-Translit-Slovene.kmap Russian-Translit.kmap Russian-extended.kmap Russian.kmap SAMPA.kmap SGML.kmap Sanskrit-Translit.kmap Sanskrit.kmap Serbian.kmap Slavic.kmap Slovak-Programmer.kmap Slovenian.kmap Spanish.kmap SpanishPrefix.kmap Syriac.kmap Tamil-Inscript.kmap Tamil.kmap TeX-smeTeX.kmap TeX.kmap Telugu-Inscript.kmap Telugu-Rts.kmap Telugu.kmap Thai.kmap Tibetan-Wylie.kmap Troff.kmap Ukrainian-Translit.kmap Urdu-ArabTeX.kmap Urdu-Nastaliq.kmap Urdu.kmap Vietnamese-TCVNcombine.kmap Vietnamese-TCVNkey.kmap Vietnamese-Telex1.kmap Vietnamese-Telex2.kmap Vietnamese.kmap Welsh.kmap 五筆字型輸入法.kmap 倉頡輸入法.kmap